Appl. Comput. Harmon. Anal. 29 (2010) 368–381

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Inpainting for compressed images

Jian-Feng Cai [a], Hui Ji [b,*], Fuchun Shang [c], Zuowei Shen [b]

[a] Department of Mathematics, University of California, Los Angeles, CA 90095, United States
[b] Department of Mathematics, National University of Singapore, Singapore 117543
[c] Temasek Laboratories and Center for Wavelets, Approximation and Information Processing, National University of Singapore, Singapore 117543

ARTICLE INFO

Article history:
Received 11 May 2009
Revised 11 January 2010
Accepted 12 January 2010
Available online 25 January 2010
Communicated by Charles K. Chui

Keywords:
Tight frame
Inpainting
Image compression

ABSTRACT

Motivated by the recent work on image inpainting in pixel domain using tight frame in J.-F. Cai et al. (2008) [3], we propose a unified iterative frame-based algorithm for inpainting compressed images to recover missing bits or missing coefficients during the compression process. Such an inpainting algorithm could be applied to any image compression scheme based on coefficients thresholding and quantization in some transform domain as a post-process to effectively remove compression artifacts, or equivalently to improve the compression ratio. The convergence of the iteration is proved and the resulted solution minimizes a special functional. Numerical experiments on popular image compression schemes demonstrate the effectiveness of our inpainting algorithm on improving the visual quality of compressed images.

1. Introduction

A wide range of applications in visual communication requires efficient image compression to fit a large amount of visual data into the narrow bandwidth of communication channels while keeping the visual quality of images acceptable. Usually part of image information has to be discarded during the compression process to achieve high compression ratio, which leads to significant degradation of visual quality of compressed images especially at a low bit rate. The visual quality of images will be greatly improved if part of the missing information during the compression process can be recovered. Such a recovery process can be described as an *inpainting* process in either coefficient domain or bit domain for compressed images.

In the digital world, the term "inpainting" [1] refers to recover lost or corrupted part of images. In recent years, many inpainting techniques [3,4,7,8] have been proposed to address various image restoration problems including reverting deterioration (e.g. cracks in photographs), removing selected elements (e.g. stamped date and text), or filling missing pieces. In this paper, we are interested in the inpainting problem related to image compression, that is, how to recover the missing information of images during compression process.

The mathematical model for such an inpainting problem is described as follows. Let $\mathbf{f} \in \mathbb{R}^N$ denote the original image by concatenating all columns of the image. Let $W \in \mathbb{R}^{N\times N}$ be the decomposition operator used in the image encoder, and let W^{-1}, the inverse of W, be the reconstruction operator in the image decoder. A typical image compression process can then be described as

$$\mathbf{c} = DW\mathbf{f}, \tag{1}$$

* Corresponding author.
E-mail addresses: cai@math.ucla.edu (J.-F. Cai), matjh@nus.edu.sg (H. Ji), tslsfc@nus.edu.sg (F. Shang), matzuows@nus.edu.sg (Z. Shen).

1063-5203/$ – see front matter
doi:10.1016/j.acha.2010.01.005

where D is a compressing operator. The compressed coefficient $\mathbf{c}$ is used in storage and transmission. The decompressed image is obtained by reconstructing the image from $\mathbf{c}$:

$$\mathbf{f}^d = W^{-1}\mathbf{c}.$$

There are two types of widely used compressing operators. One is "thresholding" operator D^t in coefficient domain defined by

$$D^t W\mathbf{f}[i] = \begin{cases} W\mathbf{f}[i], & \text{if } |W\mathbf{f}[i]| \geqslant t, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

for some threshold t. After applying the thresholding operator D^t on $W\mathbf{f}$, only those coefficients larger than t are kept, and all other small coefficients are missing. The other compressing operator is "quantization" operator D^q in bit domain defined by

$$D^q W\mathbf{f}[i] = \operatorname{sgn}\big(W\mathbf{f}[i]\big)2^q \left\lfloor \frac{|W\mathbf{f}[i]|}{2^q} \right\rfloor \tag{3}$$

for some integer q, where $\lfloor\cdot\rfloor$ is the floor operation. The parameter in (3) is in the form of 2^q since only binary-code finite decimal numbers are used in digital systems. After applying the quantization operator on each coefficient, only bits with order higher than q are kept, while the remaining lower bits are missing.[1] In practice, these two compression operators play equally important roles in optimizing the performance of image compression. It is noted that in practical de-quantization implementation, half bit will be added back to all coefficients in order to reduce reconstruction errors.

Since part of information of $\mathbf{f}$ is lost after applying the compressing operator D on $W\mathbf{f}$, the resulted decompressed image $\mathbf{f}^d$ is then only an approximation to the original image $\mathbf{f}$. The goal of inpainting for compressed images is to reconstruct a better approximation $\mathbf{f}^r$ out of $DW\mathbf{f}$ by recovering $W\mathbf{f}$ from $DW\mathbf{f}$. We propose two different inpainting processes in such a recovery process:

(i) Inpainting in bit domain: to re-fill the missing bits of all coefficients $D^q W\mathbf{f}$ with respect to $W\mathbf{f}$.
(ii) Inpainting in coefficient domain: to recover the missing coefficients of $W\mathbf{f}$ from $D^t W\mathbf{f}$.

Problem (i), inpainting in bit domain, is very interesting but also very challenging from three perspectives. First, the bit domain is the universal domain for all data in the digital world, where a very large set of analytic real numbers are represented by a relative-small set of integer values. To best of our knowledge, the mathematical study on such an important domain is very rare. It is interesting to develop analytic algorithms directly working on a discrete set. Secondly, since compressing schemes apply D^q on the bit representation of each coefficient, as a result, the compressed image only has reliable bits, but not accurate coefficients. Hence, it makes sense to recover the missing bits in bit domain. In this paper, our main focus is on the inpainting in bit domain. Finally, the inpainting process essentially is to retouch all coefficients such that the quantization residual is removed, which can be defined as a very unique denoising process where the noise is deterministic and data dependent. Regarding problem (ii), the inpainting problem in coefficient domain, it is more or less similar to those traditional image inpainting problems. They all attempt to recover the missing pieces of data and the differences among these approaches lie in the domain chosen to represent images.

Motivated by the idea in [3,4] which recovers missing pixels in $\mathbf{f}$ by finding a sparse approximation to $\mathbf{f}$ in the tight frame domain of images, we proposed a unified iterative algorithm to solve both inpainting problems for compressed images: inpainting in bit domain and inpainting in coefficient domain, or the combination thereof. In other words, the missing bits in $D^q W\mathbf{f}$ and the missing coefficients in $D^t W\mathbf{f}$ both can be recovered by our proposed algorithm. The basic idea is to improve the "guess" of the missing information in each iteration by utilizing the inherent sparseness of images in tight frame decomposition. Furthermore, we showed that the proposed algorithm is actually an iterative scheme which minimizes a special functional on the image in pixel domain together with a penalty term on the sparsity of the image in frame domain. With such an interpretation, we established the convergence of the algorithm when W is an orthonormal transform.

As one application of our inpainting algorithm in both bit domain and coefficient domain, we demonstrated how the recovered missing information from our algorithm is adequate for removing many visual artifacts in compressed images. The reduction of these artifacts results in a significant improvement on visual quality of decompressed images, which equivalently increases the compression ratio. Such a post-process on decompressed images could be very attractive in practice as it can be easily incorporated into existing industry compression techniques with few modifications. There are some post-process techniques for compressed images proposed in past to alleviate the compression artifacts (ringing, blocking, etc.) for either DCT-based compression [16,19–21] or Wavelet-based compression [17]. Unlike these ad-hoc techniques which

[1] We mean that, if the coefficient value is represented by bit (binary) format, after quantization, bits ('1's or '0's) in the positions higher than q-th order are kept intact, while bits in the q-th and lower order are set to '0's. For example, 148 in binary format is 10110100, if $q = 5$, it becomes 10100000, i.e. the last 5 bits are set to '0's.

are limited to specific type of artifacts and specific compression schemes, the post-process derived from our inpainting algorithm leads to a universal artifacts remover regardless the choice of encoding schemes and the type of induced artifacts.

The paper is organized as follows. The unified iterative algorithm for inpainting in both bit domain and coefficient domain is given in Section 2. In Section 3, we establish the equivalence of our algorithm to a special minimization and prove the convergence of the algorithm. In Section 4, we apply our algorithm on sample images compressed by both DCT-based compression (JPEG) and wavelet-based compression (JPEG2000) to evaluate the effectiveness of our inpainting post-process on improving the visual quality of compressed images.

2. Inpainting in both bit domain and coefficient domain

In this section, we present our unified inpainting algorithm in both bit domain and coefficient domain by using tight frames. Recall that the reconstructed image $\mathbf{f}^d$ without inpainting is only an approximation to the original image $\mathbf{f}$:

$$\mathbf{f}^d = W^{-1} D W \mathbf{f}.$$

Our goal is to get a better recovery $\mathbf{f}^r$ than $\mathbf{f}^d$ from the compressed coefficients $\mathbf{c} := DW\mathbf{f}$. Since the transform W is invertible and its inverse W^{-1} is uniformly bounded (e.g. wavelet transform). Recovering $\mathbf{f}$ is equivalent to recovering the wavelet coefficients $W\mathbf{f}$ from $\mathbf{c}$.

We first focus on solving the problem of inpainting in bit domain, where $D = D^q$ is the quantization operator defined in (3). Then we will show how the inpainting algorithm in bit domain can be adapted to solve the inpainting problem in coefficient domain with little modification. The goal of inpainting in bit domain is to refill the missing lower order bits of all coefficients $D^q W\mathbf{f}$ with respect to $W\mathbf{f}$.

Our algorithm is based on the tight frame systems. For a given matrix $A \in \mathbb{R}^{K\times N}$, its rows form a tight frame in $\mathbb{R}^N$ if and only if the perfect reconstruction formula $\mathbf{x} = \sum_{\mathbf{y}\in A}\langle \mathbf{x}, \mathbf{y}\rangle \mathbf{y}$ holds for all $\mathbf{x} \in \mathbb{R}^N$, or equivalently, for all $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\|^2 = \sum_{\mathbf{y}\in A} |\langle \mathbf{x}, \mathbf{y}\rangle|^2$. Notice that all rows are normalized in a tight frame system. In the matrix form, A is a tight frame if and only if $A^T A = I$. Unlike the orthonormal case, we emphasize that $AA^T \neq I$ in general. The tight frame A (rows of A) used in our implementation is derived from a tight framelet system with associated masks h_i, $i = 0, \ldots, r$ derived the unitary extension principle of [18]. The definition of the tight frame used here is coincides with that of in literature, e.g. [12,18]. The framelet decomposition and reconstruction algorithm is described in detail in [12]. In fact, the matrix A can also be viewed as the framelet decomposition operator in $\mathbb{R}^N$, and A^T is its corresponding reconstruction operator. We omit the detailed discussion here, as the detailed discussions are given in several earlier papers, e.g. [3,6,9]. It is noted that the algorithm will work for tight frame systems as long as there exists a sparse approximation of the underlying image in the chosen tight frame system. One may use tight framelet or curvelet for sparse approximation of piecewise smooth images and local DCT for sparse approximation of periodic patten as pointed out in [13]. The framelet used in the implementation of this paper is the tight framelet generated from cubic spline constructed by [18] via the unitary extension principle of [18].

Motivated by the inpainting algorithm in pixel domain [3], we propose an approach to do inpainting in bit domain by utilizing the sparseness of images in tight framelet domain. Our proposed algorithm is to seek a recovered image $\mathbf{f}^r$ such that it meets the following two requirements:

(a) *The sparsity of* $\mathbf{f}^r$ *in tight frame domain.* If real images have sparse approximations under tight frame A, the restored images $\mathbf{f}^r$ will also have a sparse approximation under A.
(b) *The data fidelity of* $\mathbf{f}^r$. If the original image $\mathbf{f}$ satisfies the model $\mathbf{c} = D^q W\mathbf{f}$, the restored image $\mathbf{f}^r$ will do the same, i.e., $D^q W\mathbf{f}^r = \mathbf{c}$. This assumption implies that $W\mathbf{f}^r \in (D^q)^{-1}\mathbf{c}$, where $(D^q)^{-1}\mathbf{c}$ is pre-image of $\mathbf{c}$ under the mapping D^q.

Let $\Lambda_0 = \{i\colon \mathbf{c}(i) = 0\}$, $\Lambda_- = \{i\colon \mathbf{c}(i) < 0\}$, and $\Lambda_+ = \{i\colon \mathbf{c}(i) > 0\}$. By the definition of D^q, the pre-image $\mathcal{S}^q$ (the feasible set of $W\mathbf{f}$) such that $D^q\mathcal{S}^q = \mathbf{c}$ is given by

$$\mathcal{S}^q := \left\{ \tilde{\mathbf{c}} \in \mathbb{R}^N\colon \tilde{\mathbf{c}}(i) \in \begin{cases} [-2^q, 2^q], & \text{if } i \in \Lambda_0, \\ [\mathbf{c}(i), \mathbf{c}(i) + 2^q], & \text{if } i \in \Lambda_+, \\ [\mathbf{c}(i) - 2^q, \mathbf{c}(i)], & \text{if } i \in \Lambda_- \end{cases} \right\}.$$

We propose to use the following iterations:

$$\mathbf{f}_{n+1} = W^{-1} P_{\mathcal{S}^q} W A^T T_\lambda A \mathbf{f}_n \tag{4}$$

to obtain $\mathbf{f}^r$. In the following discussion, we show that the iterations (4) will converge to a solution $\mathbf{f}^r$ which satisfies these two requirements (a) and (b). The implementation of (4) includes the following two-step procedure:

1. *Perform soft shrinkage on frame domain to satisfy* (a). Let A be a tight frame. We set

$$\tilde{\mathbf{f}}_n = A^T T_\lambda A \mathbf{f}_n,$$

where T_λ is the soft-thresholding operator defined by

$$T_\lambda\big([\beta_1, \beta_2, \ldots, \beta_K]^T\big) := \big[t_{\lambda_1}(\beta_1), t_{\lambda_2}(\beta_2), \ldots, t_{\lambda_K}(\beta_K)\big]^T \tag{5}$$

with $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_K]^T$, and $t_{\lambda_i}(\cdot)$ is the soft-thresholding function

$$t_{\lambda_i}(\beta_i) := \begin{cases} \operatorname{sgn}(\beta_i)(|\beta_i| - \lambda_i), & \text{if } |\beta_i| > \lambda_i, \\ 0, & \text{if } |\beta_i| \leqslant \lambda_i. \end{cases}$$

Then $\tilde{\mathbf{f}}_n$ is synthesized by a tight frame coefficient vector $T_\lambda A\mathbf{f}_n$. Notice that $T_\lambda A\mathbf{f}_n$ is a sparse vector based on the definition of T_λ. Thus, $\tilde{\mathbf{f}}_n$ has a sparse approximation under A.

2. *Project the coefficient $W\tilde{\mathbf{f}}_n$ on the feasible set $\mathcal{S}^q$ to satisfy* (b). We set

$$\mathbf{f}_{n+1} = W^{-1} P_{\mathcal{S}^q} W\tilde{\mathbf{f}}_n,$$

where the projection $P_{\mathcal{S}^q} W\tilde{\mathbf{f}}_n$ is defined by

$$P_{\mathcal{S}^q} W\tilde{\mathbf{f}}_n = \arg\min_{\mathbf{d}\in\mathcal{S}^q} \left\{ \frac{1}{2}\|\mathbf{d} - W\tilde{\mathbf{f}}_n\|_2^2 \right\}$$

and has an explicit solution:

$$[P_{\mathcal{S}^q} W\tilde{\mathbf{f}}_n](i) = \begin{cases} \max\{-2^q, \min\{[W\tilde{\mathbf{f}}_n](i), 2^q\}\}, & \text{if } i \in \Lambda_0, \\ \max\{\mathbf{c}(i), \min\{[W\tilde{\mathbf{f}}_n](i), \mathbf{c}(i) + 2^q\}\}, & \text{if } i \in \Lambda_+, \\ \max\{\mathbf{c}(i) - 2^q, \min\{[W\tilde{\mathbf{f}}_n](i), \mathbf{c}(i)\}\}, & \text{if } i \in \Lambda_-. \end{cases} \tag{6}$$

Since W is invertible, we have that $W\mathbf{f}_{n+1} = P_{\mathcal{S}^q} W\tilde{\mathbf{f}}_n \in \mathcal{S}^q$. This means that $D^q W\mathbf{f}_{n+1} = \mathbf{c}$. Therefore, the given higher order bits are replaced by the known data $\mathbf{c}$, while the missing lower order bits are obtained from the output of step 1 with some constraints.

Here we give a brief explanation of the algorithm and the quantitative analysis will be given in the next section. It is seen that all available coefficients $\mathbf{c} = D^q W\mathbf{f}$ of the image in coefficient domain are inaccurate due to quantization. Thus, the resulted image $\mathbf{f}^d = W^{-1}\mathbf{c}$ in pixel domain is not accurate either. By representing the resulted image $\mathbf{f}^d$ using redundant tight frame, the soft-shrinkage on the frame coefficients $A^T T_\lambda A\mathbf{f}^d$ leads to the perturbations among frame coefficients. Indeed, $AA^T T_\lambda A\mathbf{f}^d \neq T_\lambda A\mathbf{f}^d$ by $AA^T \neq I$ when the rows of A form a truly redundant tight frame. Thus, $AA^T T_\lambda A\mathbf{f}^d$ perturbs $A\mathbf{f}^d$. On the contrary, if rows of A form an orthonormal basis, $AA^T T_\lambda A\mathbf{f}^d = T_\lambda A\mathbf{f}^d$. Then the known coefficients do not "flow" to unknown coefficients. Therefore the redundancy in tight frame system A is very important in this approach because only thresholding in a redundant system allows the perturbations among coefficients. This explains why we need to use tight frame A to fill missing information of images instead of directly working on the coefficient domain W as rows of W form an orthonormal basis. Hence, step 1 will introduce desired perturbations which help refilling missing information based on available information. Also, as pointed out in [3,5], the redundancy of tight frames helps removing artifacts in images due to the thresholding in coefficient domain.

Step 2 is to keep the original information of $\mathbf{f}$ untouched after introducing perturbations in step 1. For perturbed coefficients from step 1, denoted by $\mathbf{d} \in \mathbb{R}^N$, $P_{\mathcal{S}^q}\mathbf{d}$ is used as the input of the next iteration. By the definition of $P_{\mathcal{S}^q}$, we have that $P_{\mathcal{S}^q}\mathbf{d} \in \mathcal{S}^q$. This, together with the definition of $\mathcal{S}^q$, implies that $D^q P_{\mathcal{S}^q}\mathbf{d} = \mathbf{c}$. Therefore, on the one hand, the higher order bits in $\mathbf{d}$ are replaced by the given data $\mathbf{c}$ such that the original information of $\mathbf{f}$ is untouched. On the other hand, the missing lower order bits are obtained from $\mathbf{d}$ with the following strategy. If $\mathbf{d}(i)$ can be quantized to $\mathbf{c}(i)$, $\mathbf{d}(i)$ is likely to be a correct restoration. Thus, we use the lower order bits of $\mathbf{d}(i)$ to fill the missing lower order bits of Wf. If $\mathbf{d}(i)$ cannot be quantized to $\mathbf{c}(i)$, $\mathbf{d}(i)$ is unlikely to be correct. Thus, we should not use the lower order bits of $\mathbf{d}(i)$ to fill the missing lower order bits. Instead, we use the bits of all ones or all zeros to fill the missing lower order bits such that $[P_{\mathcal{S}}\mathbf{d}](i)$ is as close to the $\mathbf{d}(i)$ as possible.

The algorithm for inpainting in bit domain can be modified to solve the inpainting problem in coefficient domain. For the inpainting problem in coefficient domain, the compressing operator $D = D^t$ is the thresholding operator in (2). The goal now is to recover the missing coefficients of $W\mathbf{f}$ from $D^t W\mathbf{f}$. By the definition of D^t, the pre-image $(D^t)^{-1}\mathbf{c}$, or the feasible set of $W\mathbf{f}$, becomes

$$\mathcal{S}^t := \left(D^t\right)^{-1}\mathbf{c} = \left\{ \tilde{\mathbf{c}} \in \mathbb{R}^N \colon \tilde{\mathbf{c}}(i) \in \begin{cases} \{\mathbf{c}(i)\}, & \text{if } i \in \Lambda_- \cup \Lambda_+, \\ [-t, t], & \text{if } i \in \Lambda_0 \end{cases} \right\}.$$

The corresponding projection operator is

$$[P_{\mathcal{S}^t} W\tilde{\mathbf{f}}_n](i) = \begin{cases} \mathbf{c}(i), & \text{if } i \in \Lambda_- \cup \Lambda_+, \\ \max\{-t, \min\{[W\tilde{\mathbf{f}}_n](i), t\}\}, & \text{if } i \in \Lambda_0. \end{cases} \tag{7}$$

Similar to (4), we use the iteration

$$\mathbf{f}_{n+1} = W^{-1} P_{\mathcal{S}^t} W A^T T_\lambda A\mathbf{f}_n \tag{8}$$

for the inpainting in coefficient domain.

The main difference between (4) and (8) lies in the adoption of the different projection operators. One is $P_{\mathcal{S}^q}$ in (6); the other is $P_{\mathcal{S}^t}$ in (7). However, the underlying idea of $P_{\mathcal{S}^t}$ is similar to that of $P_{\mathcal{S}^q}$. For any $\mathbf{d} \in \mathbb{R}^N$, we have $P_{\mathcal{S}^t}\mathbf{d} \in \mathcal{S}^t$, which implies that $D^t P_{\mathcal{S}^t}\mathbf{d} = \mathbf{c}$. Therefore, the coefficients in $\Lambda_+ \cup \Lambda_-$ are replaced by the given data $\mathbf{c}$, while the missing small coefficients are also obtained from $\mathbf{d}$ using the same strategy. If $\mathbf{d}(i)$ can be thresholded to $\mathbf{c}(i)$, we use $\mathbf{d}(i)$ as the missing small coefficient. If $\mathbf{d}(i)$ cannot be thresholded to $\mathbf{c}(i)$, which implies it is unreliable, we recover the missing small coefficient by the one which can be thresholded to $\mathbf{c}(i)$ and is closest to $\mathbf{d}(i)$.

3. Analysis of the algorithm

This section is devoted to the theoretical analysis of the algorithm presented in the previous section. The algorithms in the previous section can be re-written as

$$\mathbf{f}_{n+1} = W^{-1} P_{\mathcal{S}} W A^T T_\lambda A \mathbf{f}_n, \tag{9}$$

where $\mathcal{S} = \mathcal{S}^t$ for $D = D^t$ and $\mathcal{S} = \mathcal{S}^q$ for $D = D^q$. Define

$$\alpha_n = T_\lambda A \mathbf{f}_n. \tag{10}$$

Then α_n is the tight frame coefficient vector of the n-th iteration. Iteration (9) can be rewritten via α_n as the following:

$$\alpha_{n+1} = T_\lambda A W^{-1} P_{\mathcal{S}} W A^T \alpha_n. \tag{11}$$

Here we restrict our discussion to the case of W being an orthonormal transform (e.g. a wavelet transform, or a discrete cosine transform), i.e., $W^{-1} = W^T$. In this case, since $W A^T A W^T = I$, the row vectors of $F := AW^T$ form a tight frame in $\mathbb{R}^N$. Then, (11) is similar to the pixel domain inpainting algorithm in [3]:

$$\alpha_{n+1} = T_\lambda A\big(P_\Lambda \mathbf{g} + (I - P_\Lambda) A^T \alpha_n\big), \tag{12}$$

where $\mathbf{g}$ and Λ are the given pixels and their positions respectively, P_Λ is a linear project represented by a diagonal matrix with a diagonal entry 1 if the corresponding index belongs to Λ, or 0 otherwise.

In a quick glance, it seems that there are not so many differences between (11) and (12), except that the tight frame $F = AW^T$ in (11) is replaced by A in (12) and the projection $P_{\mathcal{S}}$ in (11) is replaced by $P_{\mathcal{S}^\Lambda}$ in (12):

$$P_{\mathcal{S}^\Lambda}\mathbf{d} := P_\Lambda \mathbf{g} + (I - P_\Lambda)\mathbf{d}.$$

However, there are very subtle differences between $P_{\mathcal{S}}$ and $P_{\mathcal{S}^\Lambda}$ such that the analytic analysis on (11) is no longer an easy generalization of the analysis on (12) provided in [3].

It is seen that $P_{\mathcal{S}}$ is non-linear while $P_{\mathcal{S}^\Lambda}$ is affine. The two projections are not working in the same manner. For any given $\mathbf{d} \in \mathbb{R}^N$, if $i \in \Lambda$, i.e., the pixel is known, then $[P_{\mathcal{S}^\Lambda}\mathbf{d}](i) = \mathbf{g}(i)$, which means that the pixels of given positions are replaced by the given data $\mathbf{g}$. This functions the same as the projection $P_{\mathcal{S}}$ does. However, if $i \notin \Lambda$, i.e., the pixel is missing, then $[P_{\mathcal{S}^\Lambda}\mathbf{d}](i) = \mathbf{d}(i)$. This means that the missing pixel is replaced by $\mathbf{d}(i)$ without any constraint, no matter what it is. This is the main difference between $P_{\mathcal{S}^\Lambda}$ and $P_{\mathcal{S}}$. And it is the reason why the convergence theory of (12) cannot be easily applied on (11). Furthermore, the problem of inpainting in pixel domain (12) is a linear inverse problem, while the problem of inpainting in bit domain or in coefficient domain (11) is a non-linear one.

We will extend the theory in [3] to analyze the convergence of (11). In particular, we will prove that α_n by (11) converges to a solution of the following minimization problem:

$$\min_\alpha \left\{ \min_{\tilde{\mathbf{c}} \in \mathcal{S}} \left\{ \frac{1}{2} \left\| \tilde{\mathbf{c}} - W A^T \alpha \right\|_2^2 \right\} + \frac{1}{2} \left\| \big(I - AA^T\big)\alpha \right\|_2^2 + \left\| \operatorname{diag}(\lambda)\alpha \right\|_1 \right\}. \tag{13}$$

The role of each term in (13) is explained as follows. The first term is the distance from the recovered coefficient vector $WA^T\alpha$ to the feasible set $\mathcal{S}$, hence this term enforces the fidelity. The last term enforces the sparsity of the function in the tight frame decomposition. The middle term controls the distance between the coefficients α and the range of A, i.e. the distance to the canonical tight frame coefficients of $\mathbf{f}$, such that the (weighted) ℓ_1 norm of α is approximately linked to the Besov norm of the underlying function $\mathbf{f}$ [2,15,14]. Together with the last term, the middle term actually enforces the regularity of the underlying function $\mathbf{f}$. Therefore, we conclude that (13) balances the fidelity, regularity of the function and sparsity of the function in frame decomposition, which is exactly the minimization we are seeking for.

3.1. Proximal forward–backward splitting

In this and the following subsections, we prove that α_n in (11) converges to a solution of (13). Our proof is based on the convergence theory for the proximal forward–backward splitting (PFBS) in [11], which is also used in [3] does. The purpose of the PFBS iteration is to find a solution for

$$\min_{x \in \mathcal{H}} F(x), \quad \text{with } F(x) = F_1(x) + F_2(x),$$

where $\mathcal{H}$ is a Hilbert space, $F_1(x)$ is a lower semi-continuous, convex and proper function, and $F_2(x)$ is a convex, continuously differentiable function with a Lipshitz continuous gradient, i.e.,

$$\big\|\nabla F_2(x) - \nabla F_2(y)\big\|_{\mathcal{H}} \leqslant \frac{1}{c}\|x-y\|_{\mathcal{H}}, \quad \forall x, y \in \mathcal{H}. \tag{14}$$

The iteration is

$$x_{n+1} = \operatorname{prox}_{dF_1}\big(x_n - d\nabla F_2(x_n)\big), \tag{15}$$

where $\operatorname{prox}_{dF_1}$ is the proximal operator defined by

$$\operatorname{prox}_{dF_1}(x) = \arg\min_{y\in\mathcal{H}}\left\{\frac{1}{2}\|x-y\|_{\mathcal{H}}^2 + dF_1(y)\right\}. \tag{16}$$

The following theorem from [11] is used to prove the convergence of the iteration (15).

Theorem 1. *Suppose that $\mathcal{H}$ is finite dimensional. Assume that $F_1(x)$ is a lower semi-continuous, convex and proper function, and $F_2(x)$ is a convex, continuously differentiable function satisfying* (14)*. Let d be $0 < d < 2c$. Then the iteration* (15) *converges to a minimizer of $F(x) = F_1(x) + F_2(x)$ if it exists.*

3.2. Convergence of the iteration (9)

In this subsection, we show that α_n in (11) converges to a solution of (13) when W is an orthonormal transform. The basic idea is to reformulate (11) into (15), such that $d = 1$ and

$$F(\alpha) = \min_{\tilde{\mathbf{c}}\in\mathcal{S}}\left\{\frac{1}{2}\big\|\tilde{\mathbf{c}} - WA^T\alpha\big\|_2^2\right\} + \frac{1}{2}\big\|\big(I - AA^T\big)\alpha\big\|_2^2 + \big\|\operatorname{diag}(\lambda)\alpha\big\|_1$$

with the splitting $F(\alpha) = F_1(\alpha) + F_2(\alpha)$ being

$$F_1(\alpha) = \big\|\operatorname{diag}(\lambda)\big\|_1, \qquad F_2(\alpha) = \min_{\tilde{\mathbf{c}}\in\mathcal{S}}\left\{\frac{1}{2}\big\|\tilde{\mathbf{c}} - WA^T\alpha\big\|_2^2\right\} + \frac{1}{2}\big\|\big(I - AA^T\big)\alpha\big\|_2^2. \tag{17}$$

Lemma 2. *The iteration* (11) *with $W^{-1} = W^T$ is the same as the iteration* (15) *with $d = 1$ and F_1 and F_2 given in* (17)*.*

Proof. By Lemma 3.1 in [3], we have that

$$T_\lambda\alpha = \arg\min_{\beta\in\mathbb{R}^K}\left\{\frac{1}{2}\|\alpha-\beta\|_2^2 + \big\|\operatorname{diag}(\lambda)\beta\big\|_1\right\}. \tag{18}$$

Together with the definitions of F_1 and the proximity operator in (16), (18) implies that $T_\lambda = \operatorname{prox}_{F_1}$. By directly comparing (11) and (15) with $d = 1$, it is easy to see that we only need to show that

$$AW^{-1}P_{\mathcal{S}}WA^T\alpha_n = \alpha_n - \nabla F_2(\alpha_n). \tag{19}$$

Indeed, let the indicator function of $\mathcal{S}$ be

$$E_{\mathcal{S}}(\tilde{\mathbf{c}}) = \begin{cases} 0, & \text{if } \tilde{\mathbf{c}} \in \mathcal{S}, \\ \infty, & \text{if } \tilde{\mathbf{c}} \notin \mathcal{S}. \end{cases}$$

Let Moreau's envelope for $E_{\mathcal{S}}$ be

$$\operatorname{env}_{E_{\mathcal{S}}}(\mathbf{d}) = \min_{\tilde{\mathbf{c}}\in\mathbb{R}^N}\left\{\frac{1}{2}\|\mathbf{d}-\tilde{\mathbf{c}}\|_2^2 + E_{\mathcal{S}}(\tilde{\mathbf{c}})\right\}. \tag{20}$$

By Lemma 2.5 in [11], we have

$$\frac{\partial \operatorname{env}_{E_{\mathcal{S}}}(\mathbf{d})}{\partial \mathbf{d}} = \mathbf{d} - \operatorname{prox}_{E_{\mathcal{S}}}(\mathbf{d}) = \mathbf{d} - P_{\mathcal{S}}\mathbf{d}. \tag{21}$$

The last equality is from the definition of $P_{\mathcal{S}}$. By the definition of F_2 and (20), we have

$$F_2(\alpha) = \operatorname{env}_{E_{\mathcal{S}}}\big(WA^T\alpha\big) + \frac{1}{2}\big\|\big(I - AA^T\big)\alpha\big\|_2^2.$$

This, together with (21) and the chain rule, leads to

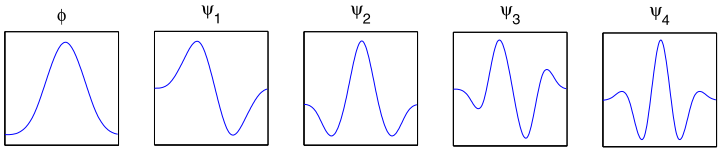


Fig. 1. Piecewise cubic framelets.

Fig. 2. Two sample images (a) "Lena" and (b) "Pepper" used in the experiments.

$$\begin{aligned}\nabla F_2(\alpha) &= AW^T \frac{\partial \mathrm{env}_{E_{\mathcal{S}}}(WA^T\alpha)}{\partial WA^T\alpha} + \big(I - AA^T\big)\alpha \\ &= AW^T\big(WA^T\alpha - P_{\mathcal{S}}WA^T\alpha\big) + \alpha - AA^T\alpha \\ &= \alpha - AW^T P_{\mathcal{S}} WA^T\alpha. \end{aligned} \tag{22}$$

Since $W^T = W^{-1}$, we obtain (19). □

Combining Theorem 1 and Lemma 2, we now have the convergence theorem for our algorithm with $W^T = W^{-1}$.

Theorem 3. *Suppose that W is an orthonormal system. Then $\alpha_n = T_\lambda A\mathbf{f}_n$, where $\mathbf{f}_n$ is generated by* (9)*, converges to a solution of* (13)*.*

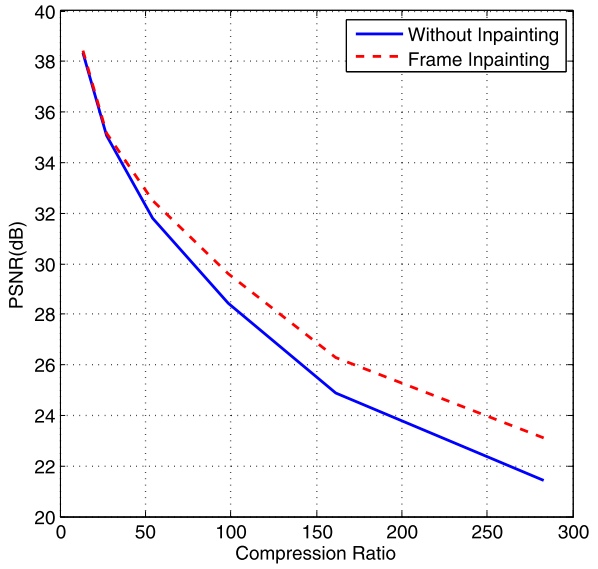
Proof. By Lemma 2, the remaining part of the proof is to check the conditions in Theorem 1. It is obvious that F_1 is proper, lower semi-continuous and convex. By Lemma 2.5 in [11], F_2 is convex and continuously differentiable, and its gradient is given in (22). Since $P_{\mathcal{S}}$ is a proximity operator, by Lemma 2.4 in [11], the operator $I - P_{\mathcal{S}}$ is non-expansive, i.e.,

$$\big\|(\tilde{\mathbf{c}} - P_{\mathcal{S}}\tilde{\mathbf{c}}) - (\mathbf{d} - P_{\mathcal{S}}\mathbf{d})\big\|_2 \leqslant \|\tilde{\mathbf{c}} - \mathbf{d}\|_2, \quad \forall \tilde{\mathbf{c}}, \mathbf{d}.$$

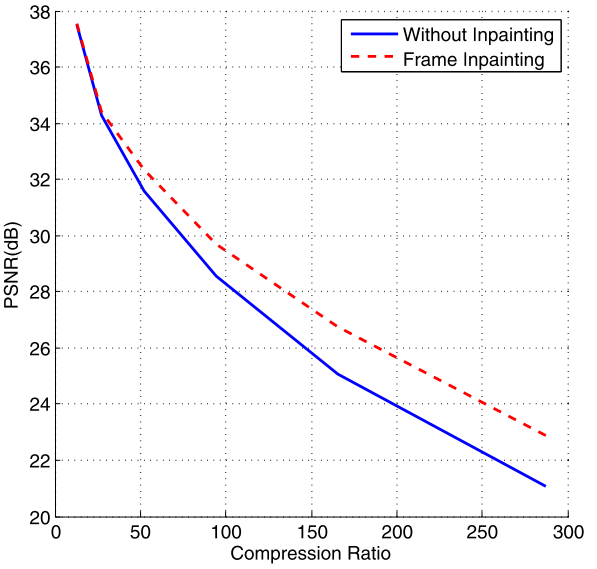
Therefore, $\forall \alpha, \beta$, we have

$$\begin{aligned}\big\|\nabla F_2(\alpha) - \nabla F_2(\beta)\big\|_2^2 &= \big\|\big(\alpha - AW^T P_{\mathcal{S}} WA^T\alpha\big) - \big(\beta - AW^T P_{\mathcal{S}} WA^T\beta\big)\big\|_2^2 \\ &= \big\|\big(I - AA^T\big)(\alpha - \beta) + AW^T\big(\big(WA^T\alpha - P_{\mathcal{S}}WA^T\alpha\big) - \big(WA^T\beta - P_{\mathcal{S}}WA^T\beta\big)\big)\big\|_2^2 \\ &= \big\|\big(I - AA^T\big)(\alpha - \beta)\big\|_2^2 + \big\|AW^T\big(\big(WA^T\alpha - P_{\mathcal{S}}WA^T\alpha\big) - \big(WA^T\beta - P_{\mathcal{S}}WA^T\beta\big)\big)\big\|_2^2 \\ &\leqslant \big\|\big(I - AA^T\big)(\alpha - \beta)\big\|_2^2 + \big\|AW^T\big\|_2^2\big\|WA^T(\alpha - \beta)\big\|_2^2 \\ &= \big\|\big(I - AA^T\big)(\alpha - \beta)\big\|_2^2 + \big\|WA^T(\alpha - \beta)\big\|_2^2 \\ &= \big\|\big(I - AA^T\big)(\alpha - \beta)\big\|_2^2 + \big\|AW^TWA^T(\alpha - \beta)\big\|_2^2 \\ &= \big\|\big(I - AA^T\big)(\alpha - \beta) + AA^T(\alpha - \beta)\big\|_2^2 \\ &= \|\alpha - \beta\|_2^2. \end{aligned}$$

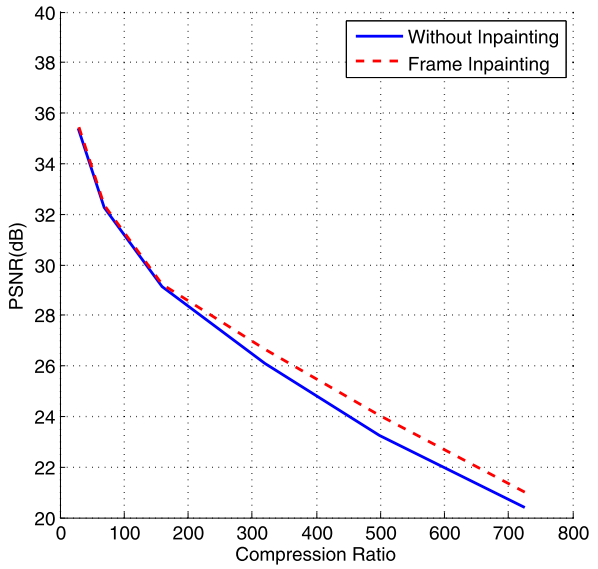
This means that F_2 satisfies (14) with $c = 1$. Furthermore, it is clear that, as $\|\alpha\|_2 \to \infty$, $\|\operatorname{diag}(\lambda)\alpha\|_1 \to \infty$. Therefore, F is coercive hence has at least one minimizer. This implies that (13) has at least one solution. Combining all together, by Theorem 1, we obtain that α_n converges to a solution of (13). □



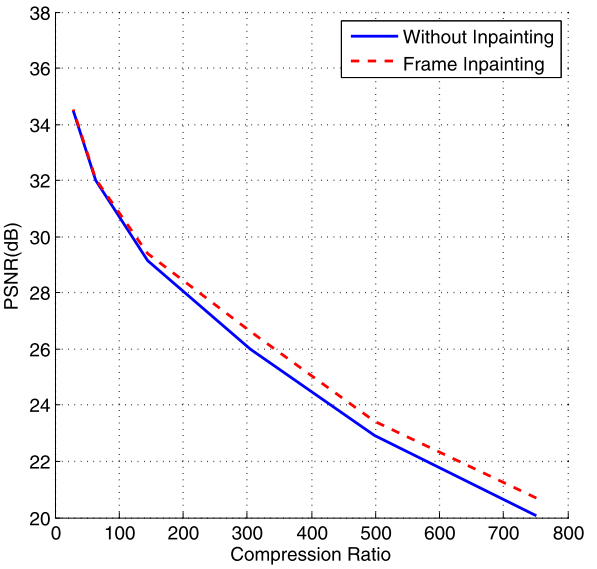
(a) DCT on Lena



(b) DCT on Pepper



(c) Bi-wavelet 9/7 on Lena



(d) Bi-wavelet 9/7 on Pepper

Fig. 3. The comparison of PSNR values of decompressed images with and without inpainting. x-axis denotes the compression ratio and y-axis denotes the PSNR of decompressed images in dB. Two compression schemes: DCT transform, and bi-orthogonal wavelet systems "9/7", are tested on two sample images "Lena" and "Pepper" shown in Fig. 2.

Theorem 3 guarantees that our proposed iterative algorithm will converge for inpainting problems in both coefficient domain and bit domain. The resulted solution is minimizing a special functional on the image $\mathbf{f}$ with desired properties.

4. Experiments

In this section, we evaluate the performance of our inpainting algorithm on removing artifacts of compressed images. In the experiments, two most popular compression schemes are chosen in the experiments: DCT-based compression and wavelet-based compression. The framelets chosen in our implementation of tight frame system is the piecewise cubic framelets (see Fig. 1) for its good balance between the support and the smoothness of its wavelet functions. More details can be found in [3]. Four levels are used in framelet decomposition. We empirically choose the threshloding parameters λ in (5) to be

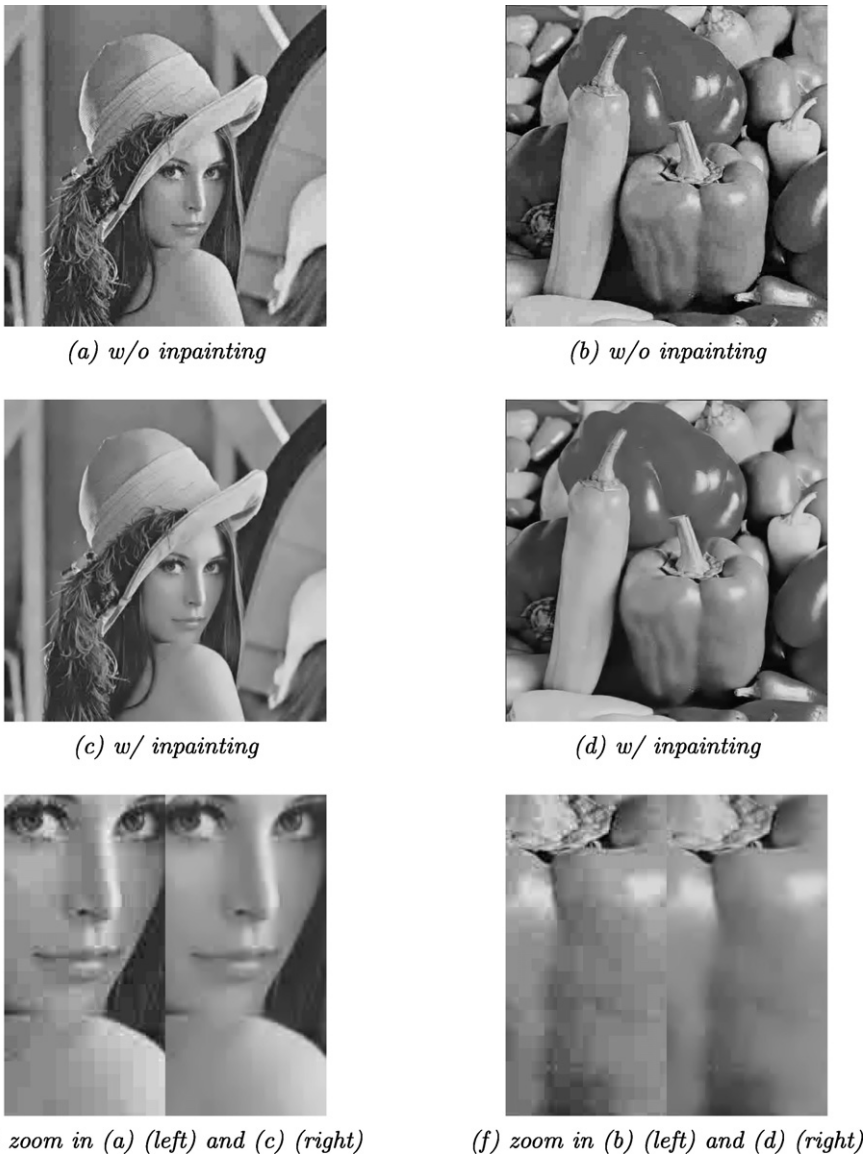


Fig. 4. The visual comparison of decompressed images compressed by DCT with and without inpainting post-process. The compression ratios are (a) 55 and (b) 54. The PSNRs of images are (a) 31.74 dB, (b) 31.48 dB, (c) 32.40 dB and (d) 32.22 dB.

$$\lambda = 3 \cdot \omega,$$

where ω is the vector of undecimated frame normalization factors and has the value $\omega_i = 2^{-\ell(i)}$ with $\ell(i)$ being the level of the index i. See [3,18] for details. It takes about 18 seconds each iteration for a 512×512 image in Matlab platform on a PC with a 2.4 GHz Intel Core 2 CPU. Depending on the compression ratio, the number of iterations ranges from 3 to 10.

We consider two different compression transforms W when compressing images. One is the discrete cosine transform (DCT); the other is the bi-orthonormal wavelet transform using 9/7 bi-orthogonal wavelet. Though compression operation D^t in the coefficient domain is theoretical sound, the remaining coefficients after thresholding are infinite decimal, which must be quantized so as to achieve the compression effect. Therefore, in practice, both thresholding operation D^t in the coefficient domain and quantization operation D^q in the bit domain are used for compressing images. During de-quantization process, half bit will be added to reduce the reconstruction error, which is a common post-processing technique used in standard compression systems. We will compare the post-process based on our inpainting algorithm in the joint domain against this widely used post-process technique.

Firstly, we want to see how the quantitative quality measurement of decompressed images can be improved by our inpainting post-process. The quality of decompressed image is measured by its peak signal-to-noise ratio (PSNR) against the original image:

(a) w/o inpainting

(b) w/o inpainting

(c) w/ inpainting

(d) w/ inpainting

(e) zoom in (a) (left) and (c) (right)

(f) zoom in (b) (left) and (d) (right)

Fig. 5. The visual comparison of decompressed images compressed by bi-orthogonal wavelets 9/7 with and without inpainting post-process. The compression ratios are (a) 319 and (b) 304. The PSNRs of images are (a) 26.04 dB, (b) 26.00 dB, (c) 26.60 dB and (d) 26.59 dB.

$$\mathrm{PSNR}(\mathbf{f}^r) = 10\log_{10}\frac{255^2}{\|\mathbf{f}-\mathbf{f}^r\|_2^2}.$$

Two sample images shown in Fig. 2 are used in this experiment. Fig. 3 shows the comparison of PSNR values of decompressed images with and without inpainting processing, with respect to different compression ratios. The compression ratio is defined as the ratio between the number of bits used before compression and after quantization for non-zero coefficients.

Fig. 3 shows that our inpainting process steadily improves the PSNR values of images compressed by both DCT and bi-orthogonal wavelet transforms. In particular, there are significant improvements on the PSNR values of de-compressed images at low bit rate (high compression ratio) when applying our inpainting algorithm on images compressed by DCT.

Figs. 4 and 5 show decompressed sample images with and without inpainting post-process for two compression schemes. It is seen that there are "block" effects and "ring" effects existing in the decompressed images without inpainting post-processing, such as the regions of girl's shoulder and the wearing hat and the body of the peppers. Most of these "block" artifacts are gone after applying our inpainting algorithm. Also, the decompressed images with inpainting process show sharper edges and visually less noisy. Such a gain in visual quality is consistent with the improvement of PSNR values of decompressed images with inpainting post-process.

In the second experiment, the frame-based inpainting method is compared against the TV (total variation) method (Model I) in [10]. The process is similar except that frame representation is replaced by TV model. Fig. 6 shows the results

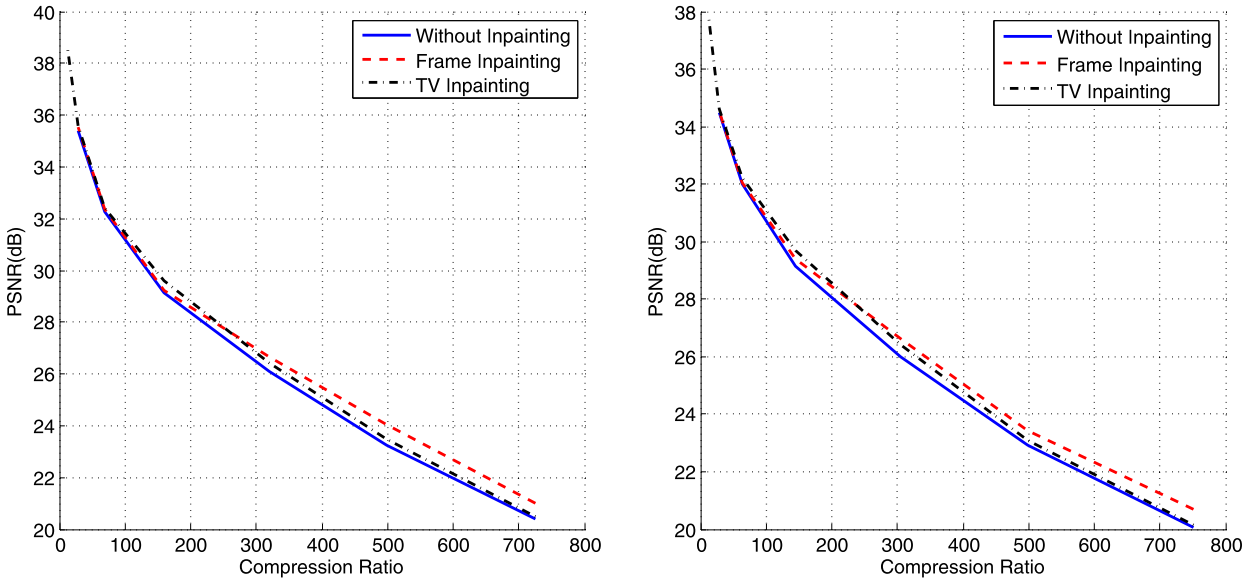


Fig. 6. Comparison of inpainting post-process based on total variation (TV) and frame for "Lena" and "Pepper".

for images compressed by wavelet transform. It is seen that when the compression ratio is high, our method noticeably outperforms TV method. However, when the compression ratio is low, TV method is slightly better. For the DCT-based compression, TV method is not a suitable method to remove block effects in compressed images, because the TV method tends to yield stair artifacts along those block artifacts in compressed images. As a comparison, our proposed framelet-based inpainting process is very effective on removing artifacts in images compressed by DCT transform. This is another advantage of our framelet method over the TV method: the framelet-based inpainting post-process can be used for both DCT based compression and wavelet based compression while the TV-based inpainting method cannot remove block effects of image compressed by DCT.

In the third experiments, the results from the frame-based inpainting method are visually compared against that from three existing methods. See Figs. 7 and 8 for the illustration on both "Lena" and "Pepper" images compressed by two methods. In additional to the TV method, we also run the method from [21] and the commercial software Topaz Vivacity [22] on the same tested images. The method from [21] performs the filtering along the block boundaries while preserving image edges in an over-complete wavelet domain. Though it is originally proposed only for de-blocking images compressed by JPEG, we find it can also be used for removing 'ring' artifacts in images compressed by wavelet-based method. It is noted that the threshold estimated in [21] does not provide optimal result. Thus, we optimized the result in terms of PSNR value by adopting the threshold T as the three times of estimated threshold for images compressed by DCT-based compression and 800 for images by wavelet-based compression. The commercial softwares from Topaz Vivacity [22] has two modules: Topaz DeJPEG for JPEG and Topaz DeJPEG2000 for JPEG2000. Topaz DeJPEG aims at eliminate JPEG artifacts (such as 'blockiness' and 'fringes') and enhance the clarity and details of the image at the same time. Topaz DeJPEG2000 aims at reducing artifacts caused by JPEG2000 and other wavelet based image compression. The results from Topaz software are optimized in terms of PSNR value by setting "suppression" parameter 76 for DeJPEG and 100 for DeJPEG2000 and choosing "best quality but slow processing" option in the software. Clearly, the frame-based inpainting algorithm produces most visually pleasant results which is also consistent with the improvement in terms of PSNR values.

5. Conclusion

In this paper, the idea of image inpainting in pixel domain using tight frame decomposition in [3,4] is generalized to the inpainting technique in either bit domain or coefficient domain. The convergence of the algorithm is also established in this paper and it is shown that the solution of the iteration minimizes a special functional with desired properties.

We would like to point out that the idea of inpainting in bit domain could possibly be foreseen a wide range of applications. Any data in digital world unavoidably suffers from the quantization error. The inpainting in bit domain provides a universal approach to improve the accuracy of the digitized data with respect to the original data. Many applications could benefit from such an accuracy gain. As one application, our proposed inpainting technique could act as a post-process on any given compression scheme to remove annoying compression artifacts, which equivalently improves the compression ratio. The experiments on two most widely used compression schemes (wavelet-based compression and DCT-based compression) justified the effectiveness of our proposed inpainting technique on improving the visual quality of compressed images.

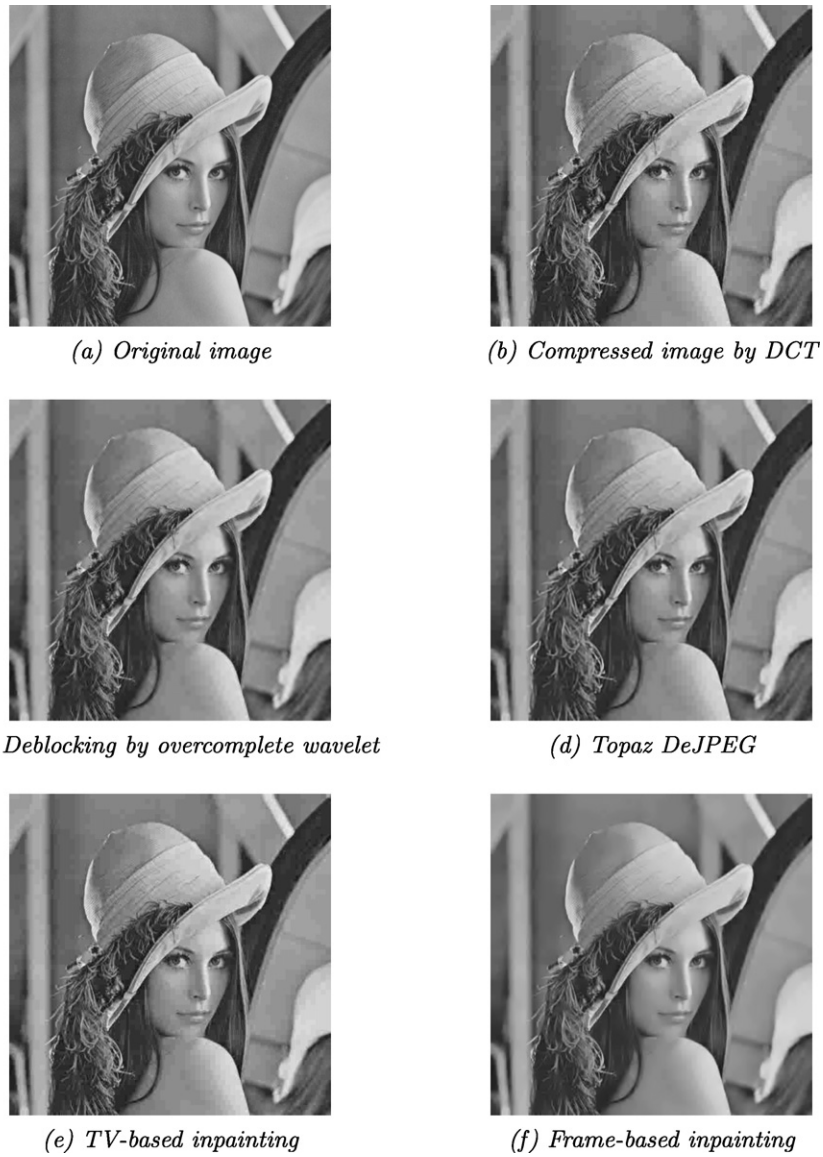
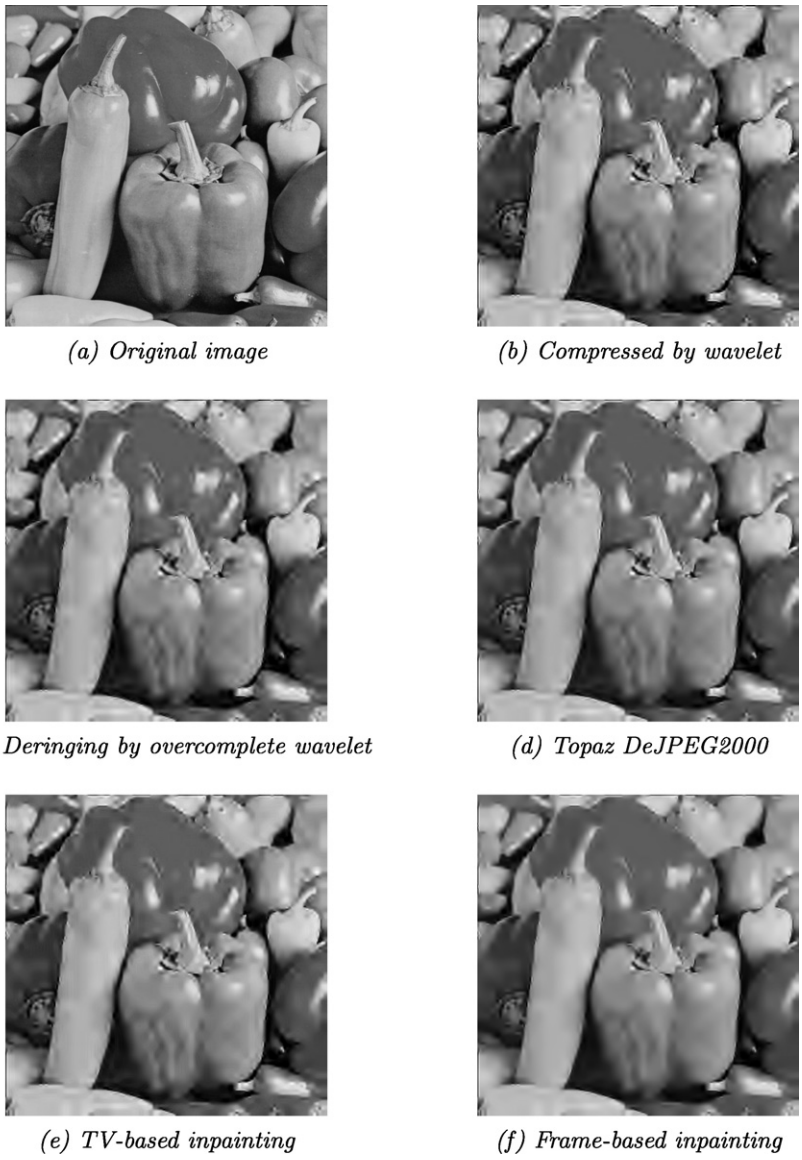


Fig. 7. The visual results of image "Lena" compressed by DCT-based method and post-processed by frame-based inpainting and other artifacts removal methods. The PSNRs of images are (b) 31.74 dB, (c) 32.36 dB, (d) 32.11 dB, (e) 31.82 dB and (f) 32.40 dB respectively.



(a) Original image (b) Compressed by wavelet

(c) Deringing by overcomplete wavelet (d) Topaz DeJPEG2000

(e) TV-based inpainting (f) Frame-based inpainting

Fig. 8. The visual results of image "Pepper" compressed by wavelet-based method and post-processed by frame-based inpainting and other artifacts removal methods. The PSNRs of images are (b) 26.00 dB, (c) 26.23 dB and (d) 26.27 dB, (e) 26.40 dB and (f) 26.59 dB respectively.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, SIGGRAPH 34 (2000) 417–424.
[2] L. Borup, R. Gribonval, M. Nielsen, Bi-framelet systems with few vanishing moments characterize Besov spaces, Appl. Comput. Harmon. Anal. 17 (2004) 3–28.
[3] J.-F. Cai, R.H. Chan, Z. Shen, A framelet-based image inpainting algorithm, Appl. Comput. Harmon. Anal. 24 (2008) 131–149.
[4] J.-F. Cai, R. Chan, L. Shen, Z. Shen, Restoration of chopped and nodded images by framelets, SIAM J. Sci. Comput. 30 (2008) 1205–1227.
[5] J.-F. Cai, R. Chan, L. Shen, Z. Shen, Convergence analysis of tight framelet approach for missing data recovery, Adv. Comput. Math. 31 (1–3) (2009) 87–113.
[6] R. Chan, S.D. Riemenschneider, L. Shen, Z. Shen, Tight frame: The efficient way for high-resolution image reconstruction, Appl. Comput. Harmon. Anal. 17 (2004) 91–115.
[7] T. Chan, J. Shen, Mathematical models for local non-texture inpainting, SIAM J. Appl. Math. 62 (2001) 1019–1043.
[8] T. Chan, J. Shen, Variational image inpainting, Comm. Pure Appl. Math. 58 (2005) 579–619.
[9] R.H. Chan, Z. Shen, T. Xia, A framelet algorithm for enhancing video stills, Appl. Comput. Harmon. Anal. 23 (2007) 153–170.
[10] T.F. Chan, J. Shen, H.-M. Zhou, Total variation wavelet inpainting, J. Math. Imaging Vision 25 (2006) 107–125.
[11] P.L. Combettes, V.R. Wajs, Signal recovery by proximal forward–backward splitting, Multiscale Model. Simul. 4 (2005) 1168–1200.
[12] I. Daubechies, B. Han, A. Ron, Z. Shen, Framelets: MRA-based constructions of wavelet frames, Appl. Comput. Harmon. Anal. 14 (2003) 1–46.
[13] M. Elad, J.-L. Starck, P. Querre, D.L. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), Appl. Comput. Harmon. Anal. 19 (2005) 340–358.

[14] B. Han, Z. Shen, Dual wavelet frames and Riesz bases in Sobolev spaces, Constr. Approx. 29 (3) (2009) 369–406.
[15] Y. Hur, A. Ron, Caplets: wavelet representations without wavelets, preprint, 2005.
[16] J. Luo, C.W. Chen, K.J. Parker, T. Huang, Artifacts reduction in low bit rate DCT-based image compression, IEEE Trans. Circuits System Video Technol. 5 (9) (1996) 1363–1368.
[17] A. Nosratinia, Postprocessing of JPEG-2000 images to remove compression artifacts, IEEE Signal Process. Lett. 10 (10) (2003) 296–299.
[18] A. Ron, Z. Shen, Affine system in $L_2(R^d)$: the analysis of the analysis operator, J. Funct. Anal. 148 (1997) 408–447.
[19] S. Wu, A. Gersho, Improved decoder for transform coding with application to the JPEG baseline system, IEEE Trans. Commun. 40 (1992) 251–254.
[20] Y. Yang, N. Galasanos, A.K. Katsaggelos, Projection-based spatially adaptive reconstruction of block-transform compressed images, IEEE Trans. Image Process. 4 (7) (1995) 896–908.
[21] Z. Xiong, M. Orchard, Y. Zhang, A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations, IEEE Trans. Circuits System Video Technol. 7 (4) (1997) 433–437.
[22] http://www.topazlabs.com/topazlabs/03products/topaz_vivacity/more_information/56tutorial_remove_jpeg_artifacts/index.html.